Recognising Re-identification Attacks on Databases, by Interpreting them as SQL Queries: a Technical Study

Olabayo Ishola[1], Eerke Boiten[1], Aladdin Ayesh[1], and Adham Albakri[1]

Cyber Technology Institute
School of Computer Science and Informatics
De Montfort University, Leicester, UK
ishola.olabayoibrahim@dmu.ac.uk
www.dmu.ac.uk/cti

Abstract. The more data sharing becomes prominent in the information age, the higher the risk of shared data being used in unexpected and undesirable ways. Data holders have employed anonymisation techniques as a means of data protection when they share a database. However, attackers can circumvent the protection or presumed protection offered by anonymisation, through re-identification attacks. Datasets are where personal information live and SQL queries are the medium through which users interact with these datasets. This paper explores from a technical perspective, how the process (killchain) of executing a re-identification attack can be represented and recognised as a series of SQL queries. Using one of the best known re-identification attack cases as a scenario, this paper explores a method for recognising re-identification attack as SQL queries on a database.

Keywords: Data Privacy · Anonymisation · Re-identification · SQL · Database Queries · Netflix Prize Data

1 Introduction

As information technology becomes more prominent in every sector of government administration, private organisation and regular human interaction, the amount of existing personal data about each individual has immensely multiplied over the years. Governments and organisations need to sometimes release information for transparency, research or Open Data projects. Example includes Unclaimed estates list and Hospital Episode Statistics (HES), these are datasets with information about individuals which could turn out to be sensitive. Research has proven that there is a risk of having sensitive personal information (for example, a medical condition of a specific person) about an individual revealed in the process [5]. The need to share data without revealing personally identifiable details brought about the concept of *anonymisation* in information sharing. Informally, a dataset is considered to be anonymised if it has been perturbed or generalised, such that it is "impossible" to attribute sensitive information to a specifically identified individual. In theory, once a database has been

"anonymised", it cannot be used to identify any specific individual or deduce any of their sensitive information from the dataset [9].

The rapid increase in the amount of publicly available information over the internet and other publicly accessible databases, coupled with advanced computer hardware and software with remarkable data processing ability has increased the possibility to analyse and "re-identify" identifiable information from an anonymised dataset. This implies that information from an anonymised dataset can be traced back to the individual to whom it relates [9].

The goal of re-identification is to turn information hoped to be anonymous into *Personal Identifiable Information* (PII), *Personal Data* or *Sensitive Data*. Even though PII is the usual target, "other information" about the data subjects which are presumed not to be personally identifiable can also be used to re-identify subjects from anonymised datasets. This "other information" is what Tore Dalenius referred to as "quasi-identifier" [13, 2].

This paper is focused on determining the technical constitution of a re-identification attack against a database, by reconstructing the SQL queries that the users of such a database might be executing. Users' access to a database is through SQL queries, so the representation of a re-identification attack process as a series of SQL queries sets a premise for exploring whether or not there is a recognisable pattern in the re-identification attack process of a database. This research reviewed the concepts associated with its research area, as presented in the next section. Through sections 3 and 4, this work employed one of the more famous re-identification attack scenarios to demonstrate possible reconstruction of a re-identification attack in the form of a series of SQL queries. This work is an attempt to practically grasp what database re-identification entails, it is an initial step towards establishing whether re-identification attack query patterns have enough in common, that they can be systematically recognised while in progress.

The overall aim of our research is to explore whether re-identification attacks can be detected in series of SQL queries. This paper explores one underlying assumption: that re-identification attacks consist of SQL queries.

2 Concepts

2.1 Anonymisation and Datasets

The need for data analysts to take full advantage of the massive availability of data in the modern information processing age, without violating the data privacy became a part of the reasons why concepts such as data anonymisation are imperative[11]. To allow data to be published and shared for research and other data processing reasons, anonymisation techniques are employed to make the dataset to be shared privacy-safe. Data anonymisation uses generalisation, perturbation, pseudonymisation and suppression techniques to present an individual's data records as indistinguishable among other records from the dataset

[1]. Anonymisation aims to prevent malicious users from inferring private or sensitive information from the dataset, however, the data should still be useful for processing by honest data analysts.

Datasets published by organisations contain quasi-identifiers. Quasi-identifiers are items of information that do not uniquely identify any individual, but can be combined with other quasi-identifiers or external, generally available data, to generate a unique identifier about an individual. De-identified (anonymised) data should not include quasi-identifiers or translation variables that allow re-identification [14].

2.2 Re-Identification and Re-Identification Attacks

Re-identification is the process whereby the data of the subjects in an anonymised dataset becomes distinguishable and can be matched with the identities of these subjects, this can occur as a result of: insufficient or poor anonymisation, the attacker having prior information about the features of a data subject and an attacker inferring sensitive details about an individual from the properties of a released dataset. The goal of anonymisation is to prevent re-identification, therefore, an attempt to perform re-identification on anonymised datasets is referred to as a re-identification attack. There are different motivations for performing re-identification by individuals and organisations. These motivations include testing the quality of anonymisation in the dataset, gaining bragging rights or professional standing for performing the re-identification, causing harm and embarrassment to the organisation that anonymised the dataset, obtaining direct benefit from the re-identified dataset, and harming or humiliating the individual whose sensitive data can be learned as a result of the re-identification [4].

It is difficult to measure the re-identification risk of a dataset, considering that the chance of a successful re-identification depends on the quality of anonymisation of the original dataset, the technical know-how of the attacker, the resources an attacker has at their disposal, and the existence of additional data that can be linked with the subjects of the anonymised dataset. Generally, the risk of re-identification will be heightened as a result of improvement in attackers' skill and techniques, increasing computational power, availability of sophisticated tools and additional information becoming available about the individuals. Computing and reporting re-identification risk likelihood will typically involve a scenario that describes the rate of success and an assumption of the attacker's resources and skill level.

Anonymised data are typically re-identified by combining two or more databases, in search of fragments of information that may reveal that the information from these databases are about the same individual. Another way re-identification of anonymised datasets can occur is by having closely related information about the same entity stored multiple times (data redundancy). When the process of re-identifying an individual from an anonymised dataset is performed successfully, it has dire data privacy protection repercussions, as it may violate the conditions under which the information was divulged by the data subject, collected and shared by the data holder.

Legally, the European data protection law GDPR [7] has a strong definition of "anonymised", requiring re-identification by anybody to be impossible. Pseudonymisation is seen as a security measure, and presumably by extension so are other weak forms of de-identification. The UK 2018 Data Protection Act [6] gives such security measures a special status by making re-identification illegal, with exemptions for research.

2.3 Objectives of Re-Identification Attacks

Different anonymised datasets offer variety of potential personal information if re-identified. Depending on the motive of the attacker, the objectives of re-identification attacks may vary. We propose the following characteristics:

Universal Re-identification: This occurs when the attacker is aiming to re-identify everyone in the dataset. The attacker in this scenario has no targets in mind, it is a penetration test on the anonymised dataset to determine the vulnerabilities in the anonymisation techniques and to explore how it can be exploited. The attack is looking to identify as many subjects as possible, therefore, its effectiveness is measured in success rate.

Existential Re-identification: This describes a re-identification attack scenario where the aim of the attack is to re-identify at least one person from the dataset, to prove that a re-identifiable subject (any subject) exists in such dataset. Re-identification is measured as either a success or a failure in this case.

Targeted Re-identification: This is a re-identification attack that focuses on a specific individual, in this scenario, the attacker already suspects that the targeted individual is in the database or may have one particular entry in the anonymised database that they want to re-identify. All re-identification strategy is geared towards identifying that specific individual. Targeted attacks are a special case of existential attacks, as the success rate of a targeted attack can be reported as an absolute yes/no or as a probability as in the Imperial College tool [15].

More generally than a full targeted re-identification attack, another outcome can also be considered; where the identity of the individual in a database is not fully discovered, but other attributes are found out. For example, a particular medical diagnosis may reveal the gender or age of a patient. This can be referred to as a re-attribution attack.

3 Experimenting with Re-identification Attacks

3.1 Case Study (Netflix Prize Dataset Case Study)

Over the years, there have been instances of successful re-identification attacks on anonymised databases, as reported in the academic literature [9, 10, 12, 16]. This has weakened the level of reliance that is placed on anonymisation techniques,

because in each case the databases were believed to be privacy-protected before they were shared publicly. This section explores one of the most significant cases of successful data re-identification in the world of data sharing. This scenario portrays a remarkable weakness in data anonymisation, when data controllers placed unjustifiable trust in their data anonymisation techniques.

This paper employs the Netflix Prize Data release as a case study for its experiment. In 2006, Netflix published one hundred million data records, divulging how hundreds of thousands of their customers had rated movies from December 1999 to December 2005. In each entry of the published record, Netflix divulged the movie rated, the rating assigned by the customers and the date the movie was rated. The data record was anonymised by Netflix before being released, anonymisation was done by removing identifying details such as usernames. However, a unique identifier was assigned to each of the users, to preserve rating-to-rating continuity (pseudonymisation). Netflix's motive for releasing this record was to use the user ratings to improve their recommendation algorithms, suggesting new movies to customers based on how favourable their ratings are to similar movies.

Weeks after the data release, Arvind Narayanan and Professor Vitaly Shmatikov[10], researchers from the University of Texas, disclosed that an attacker who has a little information from another source about a Netflix customer, can easily identify such a customer if their record is present in the dataset released by Netflix. Therefore, it is possible to re-identify individuals from the dataset with only a little outside information about their movie preferences. The researcher performed the re-identification by cross-referencing the Netflix dataset with user ratings on the IMDB (Internet Movie Database) website [9, 12].

3.2 Synthetic Data Creation

The scarcity of details regarding published successful re-identification attacks leaves room for speculation about how a particular re-identification attack could have been achieved. To begin an attempt to practically reverse engineer a re-identification attack scenario, access to technical details of the attack will be imperative.

The uncertainty around the idea that re-identification attacks may be reconstructed and understood as a series of SQL queries is the question that this work was focused on exploring. To attempt this, the Netflix Prize Data was represented by a set of synthetic data that mimics the structure and properties of the dataset published by Netflix. The synthetic data was "anonymised" using the available information about how the Netflix training dataset was scrubbed and presumed to be privacy-safe before being shared publicly [3]. The real dataset about this scenario was not used for the experiment because it would require using a real secondary dataset to complement the published Netflix training data. Although the original Netflix ratings database is still available online [8], the secondary database (IMDb) is not published in a format to be directly applicable for the purpose of this research. The synthetic dataset created for this research

work includes 3 tables. A *TrainingData* table with columns for anonymised *User ID*, *Movies*, *Date of Grade* and *Grade*; a *MovieTitles* table including columns for *Title* and the *Year* of the movie; and an *IMDb* table, that servers as a secondary data source to employ in the re-identification process. The *IMDb* table has *Username*, *Movie Name*, *Ratings* and *Date of Rating* columns.

3.3 Re-identification Attack Re-creation

Various SQL queries were executed against the dataset, with the aim of correlating a user to their movie ratings (grading) from the two different databases (the published Netflix *TrainingData* and the public Internet Movies Database, *IMDB* platform).

In an attempt to approach this systematically, a count function was executed against the *TrainingData* table, using the *User ID* column as the criterion. This is to ascertain the number of rows any specific user ID occupies. The motivation behind this strategy is to check if any of the user IDs stands out more than the others. This would make such a User ID an interesting one to explore. The synthetic dataset showed a somewhat varying output, with a particular User ID with the highest entries in the dataset. This work strategically marked the prominent User ID and other User IDs with high entries noteworthy as the experiment progressed. The result of the query is presented in Figure 1 below.

```
1  SELECT [User ID], count([User ID])
2  AS UserEntries
3  FROM [dbo].[TrainingData]
4  GROUP BY [User ID]
5  ORDER BY count([User ID])
6  DESC;
```

Results | Messages

	User ID	UserEntries
1	user 7	8
2	user 18	6
3	user 5	5
4	user 2	5
5	user 37	5
6	user 1	4
7	user 47	4
8	user 48	3
9	user 49	3
10	user 19	3
11	user 50	3
12	user 6	3
13	user 8	3
14	user 9	3
15	user 10	3
16	user 11	3
17	user 12	3

Query executed successfully.

Fig. 1. Count function with *User ID* column as criteria

The same strategy was employed using the *Movies* column as a criterion, and the returned table view showed two movies to have 14 entries; the highest number of entries. The two movies are *Black Panther* and *The Expendables*, as in Figure 2. The result of this query made two movies the points of interest at this stage.

```
SELECT [Movies], count([Movies])
AS MovieCount
FROM [dbo].[TrainingData]
GROUP BY [Movies]
ORDER BY count([Movies])
DESC;
```

After the output shown in Figure 2, the next query was targeted towards identifying which 14 users from the *TrainingData* table rated the most recurring movie, using *Black Panther* as the criterion. Now that the users that rated "Black Panther" in the *TrainingData*, alongside their rating and its date have been established, the movie *Black Panther* is used as the criterion to sort the users and their ratings in the *IMDb* table. The goal is to compare if there are any similarities in the ratings and the corresponding dates on both tables about the same movie, "Black Panther".

Results Messages

	Movies	MovieCount
1	Black Panther	14
2	The Expandebles	14
3	Source Code	12
4	Black Mirror	12
5	Dexter	12
6	Friends	11
7	How I Met Your Mother	11
8	iZombie	11
9	Spider Man	11
10	Step Up	11
11	The Big Bang Theory	10
12	Chief Daddy	10
13	Avengers	9
14	White Collor	9
15	The Pelican Brief	7
16	White Collar	1
17	The Expendebles	1

Query executed successfully.

Fig. 2. Count function with *Movies* column as criteria

```
1  SELECT *
2  FROM [dbo].[TrainingData]
3  WHERE [Movies]='Black Panther'
```

Results Messages

	User ID	Movies	Date of Grade	Grade
1	user 47	Black Panther	2004-04-22 00:00:00.000	10
2	user 19	Black Panther	2004-12-14 00:00:00.000	4
3	user 32	Black Panther	2004-04-12 00:00:00.000	2
4	user 7	Black Panther	2004-04-08 00:00:00.000	8
5	user 43	Black Panther	2004-03-01 00:00:00.000	3
6	user 20	Black Panther	2004-02-09 00:00:00.000	4
7	user 31	Black Panther	2004-07-11 00:00:00.000	2
8	user 49	Black Panther	2005-08-28 00:00:00.000	6
9	user 8	Black Panther	2005-08-11 00:00:00.000	2
10	user 2	Black Panther	2004-02-19 00:00:00.000	5
11	user 48	Black Panther	2005-01-31 00:00:00.000	2
12	user 22	Black Panther	2005-05-13 00:00:00.000	2
13	user 5	Black Panther	2006-12-09 00:00:00.000	4
14	user 1	Black Panther	2008-09-04 00:00:00.000	5

Query executed successfully.

Fig. 3. Sorting with movie *Black Panther* as criteria

At this point, there is a target criterion in focus, due to the results from preceding queries in the exploration of the *TrainingData*. Using this criterion, the movie *Blank Panther*, a query to check for its presence in the *IMDb* dataset was executed.

```
SELECT *
FROM [dbo].[IMDb]
WHERE [Movies Titles]='Black Panther'
```

Results Messages

	Username	Movie Name	Ratings	Date of Rating
1	tnoah@outlook.co.uk	Black Panther	10	2004-04-22 00:00:00.000
2	Jperalter@gmail.com	Black Panther	8	2004-12-14 00:00:00.000
3	Pnaira@gmail.com	Black Panther	4	2004-04-12 00:00:00.000
4	aliaunet@gmail.com	Black Panther	10	2005-09-12 00:00:00.000
5	cliffordharris@outlook.co.uk	Black Panther	8	2004-04-08 00:00:00.000
6	cosagie@yahoo.com	Black Panther	4	2005-07-12 00:00:00.000
7	Jmeyers@gmail.com	Black Panther	8	2005-08-09 00:00:00.000
8	Jortiz@gmail.com	Black Panther	4	2005-09-26 00:00:00.000
9	Nryan@gmail.com	Black Panther	4	2004-02-19 00:00:00.000
10	Sprokopengo@gmail.com	Black Panther	6	2005-06-06 00:00:00.000
11	Vrembrant@gmail.com	Black Panther	4	2005-05-06 00:00:00.000
12	wendyr@hotmail.co.uk	Black Panther	8	2005-08-28 00:00:00.000
13	Bross@gmail.com	Black Panther	4	2005-12-01 00:00:00.000
14	Asimbi@gmail.com	Black Panther	6	2009-06-08 00:00:00.000

Query executed successfully.

Fig. 4. *Black Panther* entries on *IMDb*

The resulting tables in Figure 3 and Figure 4 were analysed and compared for correlations between the properties (columns) of the table. Section 4 presents an analysis of this comparison, from which inferences were made about how the conclusions from the analysis are related to re-identification.

4 Analysis and Results

4.1 Analysis

The interpretation of all the information gathered from querying the datasets involved combining two tables from the two databases, and looking out for the similarities in the data that may hint that different data from different datasets is about the same individual. To attempt this, the resulting table shown in Figure 3 was joined with the one in Figure 4 for comparison. Since the movie column is already a common criteria, other columns (*Date of Grade*, *Grade*) from

TrainingData and (*Ratings*, *Date*) from *IMDb* are used for comparison between the two databases.

From the *TrainingData* table, Figure 3 shows that *User 47* rated *Black Panther* on date *2004-04-22*, with a grade of *10*. This is consistent with the entry for user *tnoah@outlook.co.uk* in the *IMDb* table as shown in Figure 4. On *2004-04-08*, *User 7* gave a rating of *8*, and on the same day, a user with username *cliffordharris@outlook.co.uk* gave the same rating to the same movie. This is also the case for *User 43* from *TrainingData* and a user *Sprokopengo@gmail.com* in the *IMDb* table.

These three entries having the same *Date of Grade/Ratings* and being rated identically is a center of attention in the query outputs generated from the exploration of the dataset with SQL. It led to a speculation that these entries from the two different datasets may be from the same individual. However, a single movie entry being identical is not a basis to conclude that these anonymous users from the Netflix Prize Data (*TrainingData*) had been re-identified in *IMDb*. To make this conclusion, a more elaborate exploration and analysis of the dataset had to be performed. For this, a *Re-identification Likelihood Quadrant* was generated, to realistically classify the probability of an entry about any particular movie being made by the same user on both platforms. The qaudrant was created based on four columns in the two databases; (*Date of Grade*, *Grade*) from *TrainingData* and (*Ratings*, *Date*) from *IMDb*.

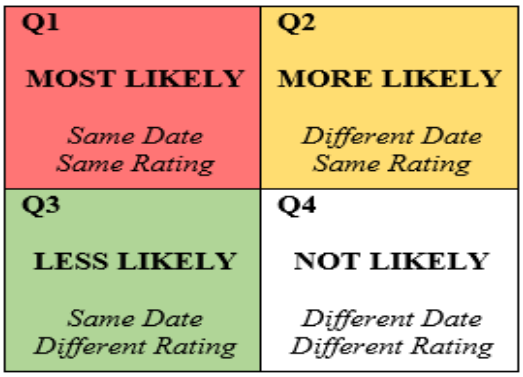

Fig. 5. Re-identification Likelihood Quadrant

Q1, in Figure 5 represents entries that were made on the same date and awarded the same rating in both datasets. These are entries that have the highest likelihood of been made by the same individuals. User(s) that falls in this quadrant can be assumed, with high probability that they have been re-identified. Q2 is for the class of users that awarded a movie the same rating, but on different dates on both datasets. There is a realistic presumption to be made about this being the same individual in both datasets, if the number of entries that

satisfy this condition is quite significant for such a user. Q2 has a likelihood of being re-dentification, depending on how prominent other determining factors are. The probability of the same person rating the same movie differently on different platforms is less likely, therefore the entries that satisfied this condition are in Q3, with very little likelihood of it being re-identification. The entries that fit the specifications of the last quadrant, presents no sign of being by the same individual, they are assumed to be ratings entered by different users.

4.2 Result

With the quadrant in Figure 5 providing a template to classify re-identification likelihood based on the Date/Rating relationship shared by the users, queries targeting each of the three users highlighted in 4.1 were executed to examine how their ratings of other movies and the date they were done classifies them into any of the sections in the re-identification likelihood quadrant.

The results of the queries for *User 47*, *User 7* and *User 43* are shown in Figures 6, 7 and 8 below. [1]

```
SELECT T.[User ID]
 ,T.[Movies]
 ,T.[Date of Grade]
 ,T.[Grade]
 ,I.[Usernames]
 ,I.[Movies Titles]
 ,I.[Ratings]
 ,I.[Date]
FROM [dbo].[TrainingData] AS T join [dbo].[IMDb] AS I ON
T.[Grade]= I.[Ratings] and
T.[Movies]= I.[Movies Titles] and
T.[Date of Grade]= I.[Date]
WHERE T.[User ID]= 'user 7'
ORDER BY [User ID]}
```

Figure 5 represents the comparison of movie ratings entries on the two databases for "User 7". The criterion of Q1 was satisfied for all of the entries made by this user. This is an indication of high likelihood of re-identification. Figure 6 shows that the criterion of Q1 was satisfied for most, but not all entries, suggesting that there is a likelihood that this is a case of re-identification. "User 43" has minimal entry that fits into Q1, re-identification likelihood is low.

From this, it was concluded with high likelihood that *User 7* from the Netflix *TrainingData* is the same individual as the user with the username *cliffordharris@outlook.co.uk*.

5 Strategy

The approach used while undertaking the re-identification in the experiment for this work was to isolate any user with properties that stood out (frequency

[1] T = TrainingData, I = IMDb

Results Messages

	User ID	Movies	Date of Grade	Grade	Username	Movie Name	Ratings	Date of Rating
1	user 7	Dexter	2004-05-03 00:00:00.000	6	cliffordharris@outlook.co.uk	Dexter	6	2004-05-03 00:00:00.000
2	user 7	Black Panther	2004-04-08 00:00:00.000	8	cliffordharris@outlook.co.uk	Black Panther	8	2004-04-08 00:00:00.000
3	user 7	Spider Man	2004-07-20 00:00:00.000	8	cliffordharris@outlook.co.uk	Spider Man	8	2004-07-20 00:00:00.000
4	user 7	Step Up	2007-08-02 00:00:00.000	7	cliffordharris@outlook.co.uk	Step Up	7	2007-08-02 00:00:00.000
5	user 7	Friends	2003-05-19 00:00:00.000	5	cliffordharris@outlook.co.uk	Friends	5	2003-05-19 00:00:00.000
6	user 7	The Big Bang Theory	2002-04-30 00:00:00.000	8	cliffordharris@outlook.co.uk	The Big Bang Theory	8	2002-04-30 00:00:00.000

Fig. 6. Query and Result for *User 7*

Results Messages

	User ID	Movies	Date of Grade	Grade	Username	Movie Name	Ratings	Date of Rating
1	user 47	Black Panther	2004-04-22 00:00:00.000	10	tnoah@outlook.co.uk	Black Panther	10	2004-04-22 00:00:00.000
2	user 47	Step Up	2005-08-09 00:00:00.000	5	tnoah@outlook.co.uk	Step Up	5	2005-08-09 00:00:00.000

Fig. 7. Result for *User 47*

Results Messages

	User ID	Movies	Date of Grade	Grade	Username	Movie Name	Ratings	Date of Rating
1	user 43	Black Mirror	2004-03-29 00:00:00.000	8	Sprokopengo@gmail.com	Black Mirror	8	2004-03-29 00:00:00.000

Fig. 8. Result for *User 43*

of occurrence in this case) the most in both the *TrainingData* and the IMDb dataset. After establishing distinctive properties from the datasets, the next step was cross-referencing these distinctive properties between the two datasets and analyse the result for re-identification clues. Recognising these properties and relating them to a secondary dataset are the two main stages involved in this strategy. The objective of the re-identification attack demonstrated in this paper is existential, as it aims directly towards proving that a user from the anonymised Netflix *TrainingData* is re-identifiable. There was no prior knowledge about any individual in the dataset, making it impractical to target any specific user.

Strategically, another re-identification objective that could also be applicable in this dataset scenario will be universal re-identification. This is because the re-identification in the dataset could be on a larger scale. However, to attempt this, a change in the attack strategy will be imperative. The strategy for a universal re-identification will focus less on making out distinctive users. Instead, efforts to cross-reference every likely output from the "count" query request will be made, thereby, re-identifying as many as possible individuals from the dataset.

Similar to Narayanan and Shmatikov in [10], re-identification success was concluded based on the likelihood that certain attributes from a secondary data source are sufficiently similar to any record in the sanitised (anonymised) database. It can be said that the work from this paper is an instance of existential re-identification, as the overall aim is to re-identify any record. The strategy for re-identification from [10] focused and depended on the use of an algorithm developed for their research work, the algorithm functioned by reading the sanitisation techniques in the *TrainingData* and the gap in the attacker's knowledge about dataset as the same. Therefore, the design of the algorithm was achieved based on the robustness of the attacker's knowledge.

6 Conclusions and future work

The work presented in this paper is only a step towards developing an automated system that monitors the user query patterns on a database and intervenes when a suspicious query is detected. There was a methodological pattern to the strategy and process used to uncover the connections between users in the two databases during the experiment for this paper. A sequence of SQL queries led to relating an individual from an anonymised database with their corresponding information in entries from another (public) database. This relation is a re-identification attack. The work presents a series of queries being executed to gradually reveal correlations that were previously unknown. The SQL queries that led to these correlations are a representation of a technical method to execute a re-identification attack against anonymised databases.

The same theory from this work will be applied to other famous re-identification cases (re-identification of Massachusetts Governor William Weld in 1997 and the AOL data breach in 2006). We will also crowd-source query data on attempted re-identification of such case studies or synthetic recreations of them, noting that for our purpose it is not even necessary that such attacks are successful. The SQL query data generated in the experiment stages for all the case studies employed, is to be used for the training of such an automated system using machine learning techniques.

References

1. The keys to data protection. Technical report, Privacy International, 62 Britton Street, London, UK, 2018. https://privacyinternational.org/sites/default/files/2018-09/Data
2. Tore Dalenius. Finding a needle in a haystack or identifying anonymous census records. *Journal of Official Statistics*, 2:329–336, 1986.
3. Dataset. Netflix prize data, https://www.kaggle.com/netflix-inc/netflix-prize-data. Online, 2017.
4. Simson L. Garfinkel. De-identification of personal information. *NIST Interagency/Internal Report (NISTIR)*, 8053, 2015.

5. Gabriel Ghinita, Panagiotis Karras, Panos Kalnis, and Nikos Mamoulis. A framework for efficient data anonymization under privacy and accuracy constraints. *ACM Trans. Database Syst.*, 34(2):9:1–9:47, July 2009.
6. Gov.uk. Data protection act 2018. *UK Public General Acts*, 2018. http://www.legislation.gov.uk/ukpga/2018/12/enacted/data.pdf.
7. ICO. Anonymisation: managing data protection risk code of practice. *Information Commissioner's Office*, 2012. https://ico.org.uk/media/1061/anonymisation-code.pdf.
8. Kaggle. Netflix prize data dataset from netflix's competition to improve their reccommendation algorithm. Date Accessed: Apr. 28, 2020. [Online]. Available: https://www.kaggle.com/netflix-inc/netflix-prize-data.
9. Boris Lubarsky. Re-identification of " anonymized data ". volume 202, 2017.https://perma.cc/86RR-JUFT.
10. A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125, May 2008.
11. Gregory S. Barnes Nelson. Practical implications of sharing data: A primer on data privacy, anonymization, and de-identification. 2015. https://www.semanticscholar.org/paper/Practical-Implications-of-Sharing-Data
12. Salvador Ochoa, Jamie Rasmussen, Christine Robson, and Michael Salib. Reidentification of individuals in Chicago's homicide database: A technical and legal study. *Massachusetts Institute of Technology*, 08 2002.
13. Paul Ohm. Broken promises of privacy: Responding to the surprising failure of anonymization. *UCLA Law Review*, 57(1701):9–12, August 2010.
14. N. Punitha and R. Amsaveni. Methods and techniques to protect the privacy information in privacy preservation data mining. *International Journal of Computer Technology and Applications (IJCTA)*, 2(6), 2011.
15. Luc Rocher, Julien Hendrickx, and Yves-Alexandre de Montjoye. Estimating the success of re-identifications in incomplete datasets using generative models. *Nature Communications*, 10(3069), 2019. https://cpg.doc.ic.ac.uk/individual-risk/.
16. Chris J Skinner and MJ Elliot. A measure of disclosure risk for microdata. *Journal of the Royal Statistical Society: series B (statistical methodology)*, 64(4):855–867, 2002.